

Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-7. (Cancelled)

Claim 8. (Currently amended) A superscalar microprocessor for processing instructions, the microprocessor comprising:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order;

a branch prediction circuit configured to provide a branch bias signal indicating whether a conditional branch controlled by a conditional branch instruction is predicted to be taken or not taken;

an instruction buffer coupled to receive fetched instructions from the instruction fetch unit and configured to buffer a plurality of fetched instructions, including an instruction selected according to the branch bias signal;

a plurality of functional units configured to execute instructions, thereby generating result data;

a register file including a plurality of entries configured to store data including result data generated by the plurality of functional units, wherein each of the plurality of entries is accessible by reference to a respective location in the register file;

a ~~decoder~~ resource identifying circuit configured to concurrently identify execution resources for a plurality of buffered instructions, thereby making a plurality of instructions concurrently available for issue, wherein the identified execution resources for each of the ~~buffered~~ available instructions ~~including~~ includes a functional unit capable of executing the instruction;

a register rename circuit configured to provide references to locations in the register file for logical register references included with the plurality of buffered instructions;

an issue control circuit coupled to the ~~decoder~~ resource identifying circuit and configured to concurrently issue more than one of the ~~decoded~~ available instructions to the functional units for execution, based on availability of the identified execution resources for each instruction and availability of respective operands for each instruction in the referenced locations in the register file, without regard to the sequential program order;

a plurality of data routing paths coupled between the plurality of functional units and the register file and configured to concurrently transfer result data from more than one of the plurality of functional units to the register file; and

bypass control logic coupled to the plurality of data routing paths and configured to ~~supply~~ distribute result data from a first one of the plurality of functional units as operand data for another one or more of the plurality of functional units via an alternate data path that bypasses the register file, wherein ~~supplying~~ distributing result data via the alternate data path occurs concurrently with transferring result data to the register file.

Claim 9. (Currently amended) The microprocessor of claim 8, wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit; and

the bypass control logic is further configured such that an integer result from the integer functional unit is ~~transferred~~ distributed to the floating-point functional unit via the alternate data path.

Claim 10. (Currently amended) The microprocessor of claim 8, wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit; and

the bypass control logic is further configured such that a floating-point result from the floating-point functional unit is ~~transferred~~ distributed to the integer functional unit via the alternate data path.

Claim 11. (Previously presented) The microprocessor of claim 8, further comprising:

operand data routing paths coupled between the register file and the functional units and configured to concurrently transfer operand data to more than one of the functional units.

Claim 12. (Previously presented) The microprocessor of claim 11, wherein the operand data routing paths transfer operand data directly from the register file to the functional units.

Claim 13. (Cancelled)

Claim 14. (Currently amended) A method for processing instructions in a superscalar microprocessor, the method comprising:

- fetching instructions from an instruction store according to a sequential program order;

- predicting whether a conditional branch controlled by a conditional branch instruction included in the fetched instructions is taken or not taken;

- buffering a plurality of fetched instructions, including an instruction selected according to the prediction, in an instruction buffer;

- concurrently identifying execution resources for more than one of a plurality of buffered instructions, the identified execution resources for each of the more than one of the plurality of buffered instructions including a functional unit capable of executing the instruction;

- providing references to locations in a register file for logical register references included with the plurality of buffered instructions, wherein the register file includes a plurality of entries, each of the plurality of entries being accessible by reference to a respective location in the register file;

concurrently making available for execution a plurality of instructions for which execution resources are identified and register file location references are provided;

concurrently issuing more than one of the plurality of available instructions ~~for which execution resources have been identified~~ for execution by a plurality of functional units, based on availability of the identified execution resources for each available instruction and availability of respective operands for each instruction in the referenced locations in the register file, without regard to the sequential program order;

executing the issued instructions in the plurality of functional units, thereby generating result data;

transferring the result data from the functional units to a the register file; and

concurrently with said act of transferring, distributing the result data from a first one of the plurality of functional units as operand data for another one or more of the plurality of functional units via a bypass data path that bypasses the register file.

15. (Previously presented) The method of claim 14, wherein:

the plurality of functional units includes an integer functional unit and a floating point functional unit; and

the act of distributing the result data includes distributing result data from the integer functional unit to the floating point functional unit via the bypass data path.

16. (Previously presented) The method of claim 14, wherein:

the plurality of functional units includes an integer functional unit and a floating point functional unit; and

the act of distributing the result data includes distributing result data from the floating point functional unit to the integer functional unit via the bypass data path.

17. (Previously presented) The method of claim 14, further comprising:

concurrently transferring operand data from the register file to more than one of the functional units via a plurality of operand data routing paths.

18. (Previously presented) The method of claim 17, wherein the operand data routing paths transfer operand data directly from the register file to the functional units.

19. (Cancelled)

Claim 20. (Previously presented) The microprocessor of claim 8, further comprising retirement control logic coupled to the register file and configured to concurrently retire a plurality of instructions according to the sequential program order.

Claim 21. (Previously presented) The microprocessor of claim 20, wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

and wherein the retirement control logic is further configured such that when an instruction is retired, corresponding result data is transferred from the temporary buffer to the retired register array.

Claim 22. (Previously presented) The method of claim 14, further comprising retiring instructions according to the sequential program order.

Claim 23. (Previously presented) The method of claim 22, wherein the register file includes:

a temporary buffer having a first plurality of entries; and
a retired register array having a second plurality of entries;
and wherein the act of retiring an instruction includes transferring corresponding result data from the temporary buffer to the retired register array.

Claim 24. (New) The microprocessor of claim 8, wherein the resource identifying circuit is further configured to concurrently identify execution resources for a first one and a second one of the plurality of buffered instructions, wherein the second one of the instructions has a data dependency on the first one of the instructions.

Claim 25. (New) The microprocessor of claim 8, wherein the register rename circuit is further configured to concurrently provide references to locations in the register file for a first and a second one of the buffered instructions, wherein the second one of

the buffered instructions has a data dependency on the first one of the buffered instructions.

Claim 26. (New) The microprocessor of claim 8, wherein the logical register references include at least one of a source register reference and a destination register reference.

Claim 27. (New) The method of claim 14, wherein concurrently identifying execution resources includes concurrently identifying execution resources for a first one and a second one of the plurality of buffered instructions, wherein the second one of the instructions has a data dependency on the first one of the instructions.

Claim 28. (New) The method of claim 14, wherein providing references to locations in the register file includes concurrently providing references to locations in the register file for at least a first one and a second one of the buffered instructions, wherein the second one of the buffered instructions has a data dependency on the first one of the buffered instructions.

Claim 29. (New) The method of claim 14, wherein the logical register references include at least one of a source register reference and a destination register reference.